Large Language Model-Based Wireless Network Design

Kehai Qiu, *Member, IEEE*, Stefanos Bakirtzis, *Member, IEEE*, Ian Wassell, Hui Song, Jie Zhang *Senior Member, IEEE*, and Kezhi Wang, *Senior Member, IEEE*

***Abstract*—In this paper, we present the Large Language Model-based combinatorial optimizer (LMCO) for wireless network optimization and planning tasks, focusing on optimizing the number and the placement of wireless access point placement. The performance and efficiency of LMCO are evaluated and compared with the well-established Ant Colony Optimization (ACO) algorithm. The results indicate that LMCO exhibits superior performance, particularly as the complexity of the problem increases. These findings also underscore the significant potential of LLM-based algorithms in revolutionizing combinatorial optimization across a wide range of applications.**

***Index Terms*—Large language model, network optimization, combinatorial optimization, access point placement**

I. Introduction

OPTIMIZATION tasks are critical in identifying the most effective solutions within a complex decision space, and they have become increasingly important in the evolving landscape of wireless communication. Moving from second-generation networks, supporting voice calls and text messages, to fifth-generation and beyond (B5G) networks, there has been a giant leap in capacity and capability. These networks are expected to provide an umbrella to the Internet of Things, machine-to-machine communications, virtual reality, and other emerging applications [1]. Commensurate progress has been observed in network planning techniques that have evolved to address increasing network complexity and the diverse needs of our ever-increasing digitalized society [2].

In wireless network planning, the traditional approach has been heavily dependent on the experience of network engineers, especially in the crucial task of selecting positions for the installation of Access Points (APs). In more recent times, their strategic decisions have been supported by radio propagation models, for example, the empirical Okumura-Hata model [3] or deterministic ray-tracing algorithms [4] to predict the received signal strength and so confirm the suitability of chosen sites.

The advent of advanced computational capabilities has shifted network planning towards an algorithmic-based approach, often supplementing or replacing human expertise with optimization algorithms. These methods, particularly metaheuristic algorithms such as evolutionary strategies [5], [6], optimize AP placement and network coverage using detailed radio propagation models. Successful network planning relies on the optimization algorithm's ability to identify optimal deployment configurations and the performance of optimization is confirmed through the use of propagation models. Recent innovations, such as optimization algorithms integrating Large Language Models (LLMs) [7]–[9], show promising results in efficiently addressing optimization issues describable using natural language. However, there is little research studying their applications in complex scenarios such as wireless network optimization, where integration with expert models is reuqired.

Against this above background, here, we aim to address the challenge of network deployment within the wireless communication sector by seamlessly incorporating an LLM-based framework with sophisticated propagation models. We introduce a *first of its kind* LLM-based optimization framework, termed Large Language Model-based combinatorial optimization (LMCO), which to our knowledge is a unique implementation in the field of wireless communications. This innovative framework demonstrates notable advantages over conventional optimization techniques. Specifically, our experiments suggest that LMCO not only surpasses traditional solutions in terms of performance, but also reveals its adaptability to address an extensive range of analogous optimization challenges.

II. Introduction of LLM and network planning modeling

A. Preliminary

A fundamental component of the LLM-based framework is the "prompt", which refers to a dynamically generated input that includes (i) the user's question, (ii) several brief examples to tune and improve the model's response, and (iii) instructions for processing the user's input.

The proposed LMCO framework aspires to replace legacy single-target optimizers, e.g., evolutionary algorithms and

Corresponding author: Kezhi Wang
This work was supported in part by the Eureka COMET (with funding from Innovate UK, No. 10099265) and Horizon Europe COVER project, No. 101086228 (with funding from UKRI grant EP/Y028031/1). The work of Stefanos Bakirtzis was supported by the Onassis Foundation and the Foundation for Education and European Culture. Kezhi would like to acknowledge the support in part by the Royal Society Industry Fellow scheme (IF-R2-23200104) and Royal Society project (IEC-NSFC-211264).
Stefanos Bakirtzis, Kehai Qiu and Ian Wassell are with the Department of Computer Science and Technology, University of Cambridge, Cambridge CB3 0FD, U.K. (kq218@cam.ac.uk; ssb45@cam.ac.uk, ijw24@cam.ac.uk), Kehai Qiu is also with Department of Computer Science, Brunel University London.
Hui Song and Jie Zhang are with Ranplan Wireless Network Design LTD, Cambridge CB23 3UY, U.K. (hui.song@ranplanwireless.com, jie.zhang@ranplanwireless.com).
Kezhi Wang is with the Department of Computer Science, Brunel University London, Middlesex UB8 3PH, UK. (kezhi.wang@brunel.ac.uk).

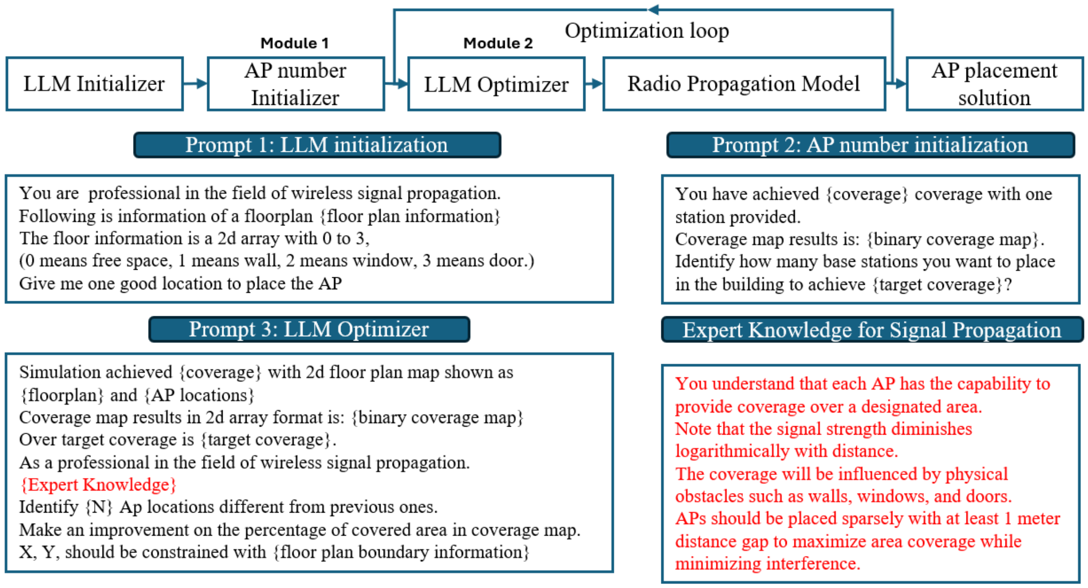

Fig. 1. The LMCO framework has 3 groups of crucial prompts that are employed to guide and instruct LMCO to solve wireless deployment optimization problems. The segments enclosed in "{}" within the prompt are placeholders that will be substituted with the relevant content when communicating with the LLM. Note that there is also a block identified with red text providing expert knowledge to the LLM. Prompt 1 is associated with LLM Initializer and Prompt 2 and Prompt 3 are associated with Module 1 and 2 respectively

ant colony optimization [10], [11] in the wireless network planning and optimization process. Unlike conventional approaches that require step-by-step programming of combinatorial optimizers, LMCO does not necessitate providing the LLM with detailed instructions on precisely executing each optimization step. Instead, it relies minimally on domain expertise supplied as domain information. Lastly, to facilitate LMCO's interpretation of the output, the prompt strictly specifies the format in which the LLM should present its results. The following subsections will outline the details of the AP placement optimization problem formulation, the structure of the LMCO framework, its inputs and outputs, as well as the implementation of its constitutive components.

B. AP Placement Task for Network Optimization

The primary objective of network planning optimization frameworks is to determine the network topology, i.e., the number of APs and their locations, that optimize and meet some target key performance indicators (KPIs), e.g., coverage, delay, power supply, or installation and administration costs [12]. In this work, we consider the optimization task of meeting a target coverage level, ϕ, while minimizing the installation costs, i.e., the number of APs, N. The corresponding optimization problem T can be formulated as follow:

$$\begin{aligned} & \min_{\{x_n,y_n\}} N \\ \text{s.t.} \quad & \sum_{k=1}^{N} R_k \geq \phi, \\ & x_{\min} \leq x_n \leq x_{\max}, \forall n \in \mathcal{N}, \\ & y_{\min} \leq y_n \leq y_{\max}, \forall n \in \mathcal{N}, \\ & \min\{|x_i - x_j|, |y_i - y_j|\} \geq L, \forall n, l \in \mathcal{N} \end{aligned} \tag{1}$$

where (x_n, y_n) indicates the location of the n-th AP, R_k is a function that evaluates the coverage for each AP and is calculated as the proportion of area where its pathloss value is lower than 90dB. We are using 3D-ray-tracing software (Ranplan Professional) [13] for coverage evaluation in this paper, but it can be replaced by other pathloss simulation tools. $x_{\min}$, $x_{\max}$, $y_{\min}$, $y_{\max}$ refer to the boundary constraints of the AP locations in a square scenario, L is the minimum distance between APs. It is set to 1 m in the following experiments, ensuring that APs do not overlap, and the constraint $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \geq L, \quad \forall i \neq j$ guarantees that the distance between any two APs is greater than or equal to L. Next, we will show how we can use the proposed LLM-based framework, i.e., LMCO to address the above problem.

III. LMCO Framework

A. LMCO

A block diagram depicting the LMCO framework is presented in Fig. 1, showing the workflow and the prototype prompts crafted for the utilization of an LLM in the context of network planning. The framework entails two modules, leveraging the in-context learning potential of the LLM, orchestrated by strategically formulated prompts: (i) an initialization and (ii) an LLM-driven deployment optimization module, whose details are presented in Algorithms 1 and 2, respectively. The initialization module is used to automatically determine the minimum number, N_{min}, of APs to be deployed, i.e., its output is N_{min}. Then, given this upper bound, the LLM-driven optimization module outputs the required number of APs, N ($N \geq \ N_{min}$), and their respective locations, s_loc, resulting in a network deployment that meets the target coverage levels.

The input for both modules is the optimization task, T, the floor plan, F, of the indoor environment where the APs will be deployed, and the target coverage level, P (note that the second module also receives as input N_{min}, provided by the first module). The floor plan comprises a grid of points depicting the wall layout, and the construction materials used, and it is represented as a two-dimensional ($2D$) array with its elements set to values depending upon the type of materials. Finally, the target coverage refers to the percentage of grid points at which the received signal strength (RSS), evaluated by R_k in (1), is larger than a threshold.

The functionality of the LLM for each module is configured through meticulously designed prompts. The selected prompts for the other two modules, along with their implementation specifics, are detailed in the following subsections. Notably, we utilize the LLM in a zero-shot setting, with diverse prompts enabling it to handle various tasks effectively.

1) Module 1: AP number Initializer: As mentioned previously, the goal of the AP number initializer is to determine the minimum number of APs to deploy without human intervention. Consequently, in the first instance, a designed prompt for the initialization module is provided to the LLM. Based on this prompt, the LLM will generate a location s_loc, where the AP will be deployed. Then, a ray tracing simulator is employed to simulate the RSS distribution in the indoor environment and calculate the coverage. This output, in turn, is utilized by LMCO to ascertain a preliminary AP count, setting the stage for subsequent optimization processes.

2) Module 2: LLM Optimizer: The second module integrates an LLM as a combinatorial optimizer, operating in a zero-shot fashion. Again, a designed prompt informs the LLM about (i) the geometry layout, (ii) the initial number of APs indicated by Module 1, (iii) the network topology, (iv) the attained coverage, through a binary coverage heatmap representation (with 0 or 1 indicating the pathloss at a certain location is below or above the threshold, respectively) and an aggregated coverage percentage. Note that the latter constitutes the objective function of (1) that the LMCO aspires to maximize. In addition, an *expert knowledge prompt* is embedded in the LLM optimizer prompt, to induce common network engineering knowledge to the LLM and orchestrate its actions. Given these pieces of information, the LLM is asked to provide the number of APs, N, and their 2D locations (different from the previous ones) such that the coverage is improved. As shown in Algorithm 2, this process is repeated iteratively until the desired coverage level is reached, and at each iteration, the RSS and the coverage are evaluated with a ray-tracer for the network deployment indicated by the LLM. Based on our observations of the LLM's optimization process, we found that after a certain number of attempts with a fixed number of APs, it stops proposing new candidate solutions, typically after six steps. Consequently, during this iterative process, if the achieved coverage does not improve after six consecutive iterations, we increase the initial number of available APs by one. This approach helps to avoid wasting computational resources.

B. Coverage Evaluation via Ray Tracing

Our target is to identify an optimized configuration of an AP locations such that when they are deployed in our target environment and the path loss is calculated using the propagation model, the resulting coverage fulfills our specified criteria. Figure 2 showcases a scenario with a floor plan and includes a path loss map derived from a particular placement of AP. We begin by setting a path loss threshold— where any location with path loss exceeding this threshold is classified as an uncovered zone. We then measure coverage as the percentage of the area that achieves satisfactory signal strength within the established path loss threshold.

Algorithm 1 AP Number Initialization algorithm

Input: Optimization task: T, floorplan: F, target coverage: P
Feedback Input: Coverage map: M, AP locations: s_loc, Coverage: s_cov
Output: Initial number of APs: N

Prompt 1 $\leftarrow$ construct prompt 1 with T, F, P.
$s\_loc \leftarrow$ instruct LLM with initialization prompt
$s\_cov \leftarrow$ set AP at s_loc in F and calculate with ray tracer.
Prompt 2 $\leftarrow$ construct prompt 2 with F, s_cov, s_loc.
Initial AP number $N \leftarrow$ instruct LLM with prompt 2
return N

Algorithm 2 LLM-driven Optimization for AP Placement

Input: Optimization task T, floorplan F, target coverage P.
Feedback Input: Coverage map M, AP count N.
Output: N optimal AP locations: s_loc.

$N \leftarrow$ get initial AP number with algorithm 1
while $s\_cov < P$ **do**
    Optimization prompt $\leftarrow$ construct prompt 3 with T, F, P, N, s_cov, s_loc.
    $s\_loc \leftarrow$ instruct LLM with optimization prompt.
    $s\_cov \leftarrow$ set AP at s_loc in F and calculate M with ray tracer.
    $N = N + 1$ if $\frac{s\_cov}{P} < 0.6$ or after 6 attempts
end while
Confirm s_cov meets P with s_loc
return s_loc

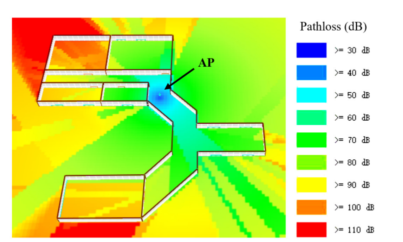

Fig. 2. A sample scenario encompasses details of the floorplan and the path loss map computed with ray-tracer

C. Baseline Metaheuristic Optimization Algorithm

The performance of LMCO will be benchmarked against ACO, a powerful and widely used metaheuristic algorithm inspired by the foraging behavior of ants, and it is particularly effective for combinatorial optimization problems.

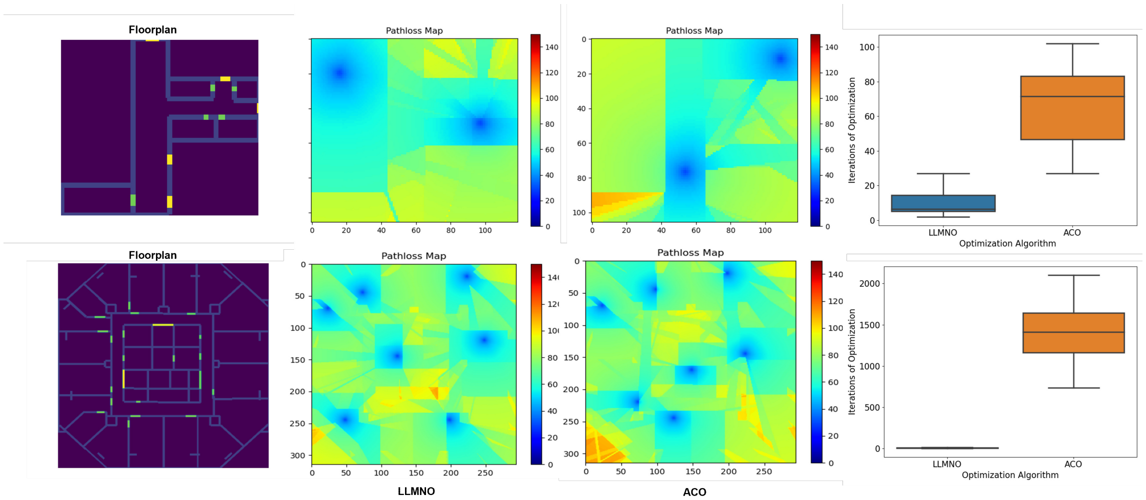

Fig. 3. Comparative Results of AP Placement in Two Buildings, the upper row building has 2 APs and the lower row building has 7 APs: Iteration Efficiency of LMCO versus ACO

In ACO, 'artificial ants' identify a nearly-optimal solution by iteratively traversing different points of the optimization space, in an analogous way that real ants use pheromone trails to identify the most efficient routes to resources. Specifically, starting from some initial random solutions, at each iteration the results of each ant are compared to determine a weight that signifies the goodness of the considered solution in a similar way that ants use pheromone to mark desirable paths. Then, new solutions are chosen probabilistically for the next iteration based on the estimated weights, i.e., the pheromone levels, and this process is repeated until the objective cost function converges. The way the weights and the path selection probability are computed depends on the implementation of the ACO algorithm. Throughout this paper, we adopt a greedy ACO implementation similar to the one discussed in [14].

In network planning, ACO uses a given number of simulated ants to investigate possible AP topologies and evaluate their coverage through ray tracing simulations. Throughout this process, the ant's decision on the number of APs and their locations is informed by the intensity of the pheromone trails that indicate the achieved coverage at each point of the optimization space. Relying on these decisions, the ants explore the optimization space, aiming at maximizing network coverage.

IV. Evaluation in practical use cases

To showcase the potential of LMCO to solve combinatorial problems and assist network planning we consider two use cases. The first experiment is conducted in a controlled environment, where a fixed number of APs, N, is predetermined and the aim is to achieve an optimal coverage metric. In the second experiment, we do not fix the number of APs and seek solutions that would meet our predefined coverage requirements. This approach allows us to evaluate the models' adaptability and efficiency in achieving the target coverage levels, which can entail different numbers of APs. In both experimental setups, our goal is to ensure that at least in 90% of the intended coverage area, i.e., $P = 0.9$, the pathloss is lower than 90dB.

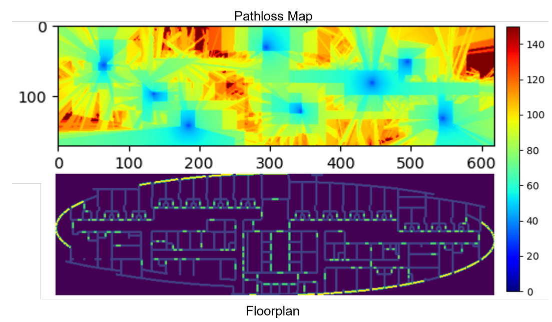

Fig. 4. Result of Real-world Scenario Optimization for AP Placement by LMCO without Prior Knowledge of the Number of APs

To ensure a fair evaluation, we design two experimental protocols, each representing a common scenario in wireless network design. These scenarios are selected to assess different aspects of performance. For each experimental setup, we conduct 20 tests using OpenAI's gpt-4-turbo-preview iteration as the LLM within the LMCO strategy. The evaporation rate ρ for the baseline ACO algorithm is set to 0.1.

LMCO is an LLM-based framework that significantly differs from existing optimization algorithms in that it lacks a quantitative representation of the algorithms used to (1) propose candidates and (2) search for subsequent candidates within the search space. Consequently, at this stage, we propose only comparing the number of iterations required by LMCO and ACO algorithms to suggest new candidates and the time taken to achieve final convergence.

1) Experiment to find a solution for a given number of APs: In this experiment, we skip the initialization phase in the LMCO algorithm due to the use of a predetermined number of APs and we conduct tests in two indoor environments. The first one assumes a simple geometric space measuring 23.8m by 20.2m and a more complex configuration with dimensions of 58.8m by 63m. In both cases, the number of APs used in each scenario was determined by prior experimentation, which established the number of APs required to meet the coverage criteria.

TABLE I
COMPARISON OF AVERAGE ITERATIONS REQUIRED FOR LMCO AND ACO IN A REAL-WORLD COMPLEX SCENARIO.

Algorithms	**LMCO**	**ACO**
Iterations	16	2275
Time used(s)	197	11802

Figure 3 illustrates the results for a fixed number of APs. In both scenarios, the LMCO strategy significantly outperforms the ACO method. On average, LMCO requires only 9.2 iterations to meet the coverage criteria in a simple scenario with 2 APs, whereas ACO requires 63.4 iterations. The average time taken by LMCO, including communication and ray-tracing simulation, is 85 s, compared to 357 s for ACO, including ray-tracing simulation time.

In the more complex scenario with 7 APs, LMCO averages 10.9 iterations and 94 s to execute, while ACO requires a substantial 1394 and 7206s respectively to achieve comparable coverage levels. Given the inherent stochastic nature of both algorithms, the results are presented as a bar plot, illustrating the distribution of iterations used across 20 independent tests for both scenarios.

2) Experiment on finding a solution without fixing the number of APs: In an advanced experiment, we did not provide either algorithm with information regarding the number of APs needed to meet coverage requirements. Figure 4 illustrates the superior configuration determined by LMCO and depicts its optimization process over a series of iterations. We executed 20 independent trials for each algorithm, incorporating an early stop mechanism. Suppose that the initial random configuration for ACO, or the prompt-based initialization for LMCO, with N APs yields a solution where only 60% of the intended coverage area satisfies the maximum threshold of pathloss of 90dB. In this case, the algorithm proceeds to test the scenario with $N+1$ APs. After the initial setup, both algorithms iteratively optimize the positioning of the APs under the current fixed number of APs. This optimization process continues until the coverage criterion is met. If the coverage criterion cannot be satisfied with the existing number of APs, the algorithm increases the number of APs by one and resumes the optimization process. This iterative process of adjusting the AP locations and the number of APs is repeated until the coverage criterion is successfully met.

Table 1 presents the average number of iterations and time required by both optimization algorithms in this scenario, and the results indicate a clear difference in performance between the LMCO and ACO algorithms for AP placement.

V. CONCLUSION AND FUTURE WORK

The result shows that the LMCO algorithm consistently required fewer iterations to achieve the desired coverage criterion across both simple and complex environments, and the bar plot in Figure 3 further supports the robust performance of LMCO over ACO, reinforcing the former's reliability and potential as a practical solution for network design. Moreover, an experiment in a real-world scenario, as depicted in Figure 4, provides evidence of LMCO's adaptability and further confirms its robustness.

In conclusion, LMCO introduces a novel framework for LLM-based optimization in wireless communications. It presents a flexible and generalized LLM-based optimizer that incorporates expert knowledge, enabling its application across domains to address increasingly complex problems. The LMCO algorithm demonstrates significant improvements in iteration efficiency and robustness, which are crucial for the requirements of large-scale wireless network deployments and real-time applications. The comparison highlights LMCO's superiority in handling complex and dynamic network configurations, substantially reducing time and computational overhead. This establishes LMCO as the preferred choice for optimizing wireless communication systems, with potential benefits extending to other optimization scenarios. Continuing this work will enhance the existing capabilities of LMCO within wireless network optimization and explore its scalability and adaptability to challenges across diverse domains.

REFERENCES

[1] M. Agiwal *et al.*, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.

[2] D. Soldani *et al.*, "Horizon 2020 and beyond: On the 5g operating system for a true digital society," *IEEE Vehicular Technology Magazine*, vol. 10, no. 1, pp. 32–42, 2015.

[3] Y. Singh, "Comparison of okumura, hata and cost-231 models on the basis of path loss and signal strength," *International journal of computer applications*, vol. 59, no. 11, 2012.

[4] Z. Yun *et al.*, "Ray tracing for radio propagation modeling: Principles and applications," *IEEE access*, vol. 3, pp. 1089–1100, 2015.

[5] Z. A. Dahi *et al.*, "Models and solvers for coverage optimisation in cellular networks: Review and analysis," in *2022 IEEE 9th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, IEEE, 2022, pp. 312–319.

[6] N. Lakshminarasimman *et al.*, "Evolutionary multiobjective optimization of cellular base station locations using modified nsga-ii," *Wireless Networks*, vol. 17, pp. 597–609, 2011.

[7] C. Yang *et al.*, "Large language models as optimizers," *arXiv preprint arXiv:2309.03409*, 2023.

[8] S. Liu *et al.*, "Large language models as evolutionary optimizers," *arXiv preprint arXiv:2310.19046*, 2023.

[9] T. Liu *et al.*, "Large language models to enhance bayesian optimization," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=OOxotBmGol.

[10] D. Simon, *Evolutionary optimization algorithms*. John Wiley & Sons, 2013.

[11] M. Dorigo *et al.*, *Ant colony optimization: overview and recent advances*. Springer, 2019.

[12] Y. Lee *et al.*, "Optimization of ap placement and channel assignment in wireless lans," in *27th Annual IEEE Conference on Local Computer Networks, 2002. Proceedings. LCN 2002.*, IEEE, 2002, pp. 831–836.

[13] Ranplan Wireless Network Design, *Ranplan professional*, https://www.ranplanwireless.com/gb/products/professional/, accessed 2023.

[14] X. Liu *et al.*, "Ant colony optimization with greedy migration mechanism for node deployment in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 39, pp. 310–318, 2014.